Rocket® software

COBOL 統合開発環境製品

Rocket® Visual COBOL™

(formerly a Micro Focus® product)

Visual COBOL は、最新版の COBOL 統合開発環境製品です。最新テクノロジー環境において、様々なミドルウェア等と連携した Windows/Linux/UNIX ターゲットの COBOL アプリケーション開発を可能にします。本製品に搭載された強力な COBOL のコーディング支援機能、デバッグ機能、COBOL 専用の開発補助機能を活用することで高い生産性が見込めます。個々の機能は様々なシーンでの使用を想定して設計されており、CI ツール等と連携させることによって変化に強い高速な開発スタイルを COBOL アプリケーション開発で実現することも可能です。

概要

Visual COBOL は、価値ある既存 COBOL 資産の最新テクノロジー環境での活用を可能にする製品です。単一の COBOL ソースを多数のプラットフォームに展開することができるため、開発期間の短縮やコスト削減にも貢献します。

開発環境として Visual Studio や Visual Studio Code および Eclipse IDE の利用が可能で、COBOL を熟知した開発者にもCOBOL が初めての開発者にも高い生産性をもたらします。Visual COBOL の開発環境／実行環境は、Windows および主要 Linux/UNIX に対応し、AWS や Microsoft Azure などクラウド環境へのCOBOL アプリケーションのデプロイも可能です。

Visual COBOL には「JVM COBOL」が搭載されており、COBOL アプリケーションを Java クラスとして動作可能にします。COBOL で記述したユーザーアプリケーションが Java のバイトコードとして Java 仮想マシンの上で動くため、業界標準のデータセンター運用に準拠したアプリケーション展開を実現します。

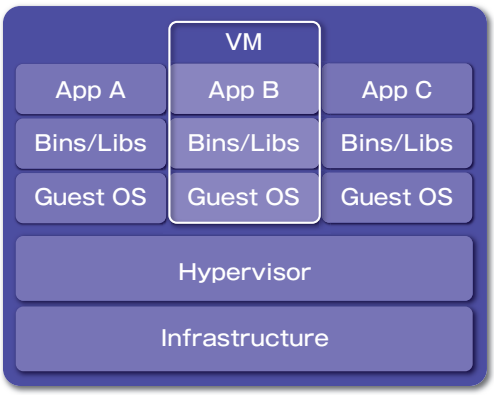
加えて、コンテナ型の仮想環境である Docker と Podman コンテナにも対応しています。

物理サーバー上で複数の OS を稼働させる仮想化はサーバー資源の効率化を実現しますが、現在主流のハイパーバイザー型の仮想化は仮想環境（仮想マシン）毎にゲスト OS が必要です。

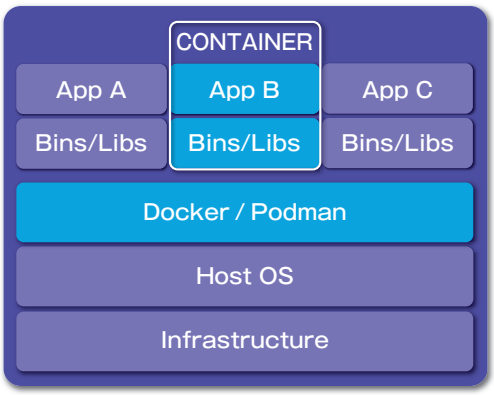
一方、コンテナ型の仮想化は、Linux カーネルの「コンテナ」機能を使って実行環境を他のプロセスから隔離し、その中でアプリケーションを動作させ、ホスト OS を共有します。コンテナのオーバーヘッドはプロセスとほぼ同一で、仮想マシンのオーバーヘッドと比較すると非常に軽量です。また、コンテナはアプリケーションとライブラリを同一のコンテナ内に固めるため、アプリケーションの移動やデプロイが容易です。

開発/テスト工程ではトラブル発生時の開発環境の再構築や大規模開発における開発環境の展開を実現し、本番運用では検証済み実行環境から本番環境を構築可能です。

アプリケーションレベルの仮想化イメージ



コンテナレベルの仮想化イメージ



製品の特長

最先端のCOBOLコンパイラ

› 64bit ネイティブコードと .NET Framework 4.x 対応の 64bit マネージドコード、検証可能コードをフルサポート
64bit 環境で 64bit/32bit アプリケーションの開発が可能
注）OS による 32bit 開発のサポートが必要です

› 国際標準 COBOL2002 規格の主要機能をサポート
- オブジェクト指向機能
- 翻訳指令
- 利用者定義データ型
- 利用者定義関数
- 再帰呼び出し
- 局所場所節
- ブール演算子
- 自由形式のソースプログラムや登録集原文
- テーブルソート
- ファイル共有・排他
- 31 桁数字項目
- EXIT PERFORM 文
- PIC 1 ビットデータ項目

› 旧 COBOL85 規格（組み込み関数も含む）上位水準にも準拠

› JVM、.NET アプリケーション記述のための拡張されたオブジェクト指向構文
- TRY/CATCHによる例外捕捉
- 手続き部中の局所変数宣言
- インターフェイス、デリゲート、列挙型、ValueType、Operator の定義
- アクセシビリティ記述 PRIVATE/PUBLIC/PROTECTED/INTERNAL/PROTECTED INTERNAL
- パーシャルクラス、ジェネリッククラスをCOBOL構文でサポート
- オブジェクト名のインライン記述（引用符で括らない参照）
- Java, C# 等との差分を吸収する拡張構文(ATTACH 文、DETACH 文、RESET collection 文等)

› グローバルアプリケーションのための UNICODE サポート

› メインフレーム COBOL 方言との互換性
- OS/VS COBOL、DOS/VS COBOL、VS COBOL II拡張構文
- Enterprise COBOL、COBOL/370拡張構文
- 日立･富士通方言のCOPY文法

› 従来の Rocket COBOL 方言との互換性

› ACUCOBOL 方言との互換性
- 環境変数アクセス構文
- 画面節の追加構文
- C$ ライブラリルーチン
- Vision ファイルシステム

› 既存資産のリプラットフォームを容易にするさまざまな独自拡張構文をサポート

› スタンドアロン PC からエンタープライズサーバーに至るまでのスケーラビリティを実現

› コンパイラ指令確定機能
- SQL 指令のない EXEC SQL 構文
- 上書き指令の確認、リセット

Java および Java EE との COBOL の使用

Visual COBOLのCOBOL/Java相互運用性のサポートにより、COBOLと Java の混成アプリケーションも、プログラマによる複雑で環境に依存したインターフェイスのコーディングなしに作成することが可能となり、また COBOL 資産内に記述されているビジネスロジックが Java および Java EE アプリケーションに拡張されます。

Interface Mapping Toolkit によって EJB が作成され、、COBOL Server および Java EE Connector との連動によって展開されると、WebSphere や WebLogic、JBoss といった主要なJava アプリケーションサーバーからの、Java EE 準拠形式での COBOL 資産へのアクセスが可能になります。

Direct COBOL Web サービス

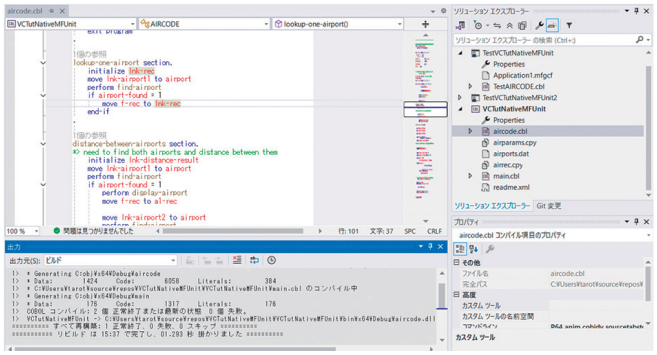
Visual COBOL の Direct COBOL Web サービス機能により、サードパーティのアプリケーションを必要とせずに SOAP に準拠した Web サービス並びに RESTful な Web サービスを作成、利用および展開することができます。Interface Mapping Toolkit は COBOL リンケージセクション中の COBOL のデータ型を SOAP もしくは JSON の適切な型へマッピングします。SOAP の Web サービスとして公開するのに必要な WSDL や RESTful として呼び出すのに必要な情報を記述した JSON ファイルも自動生成されます。

Interface Mapping Toolkit で作成した Direct COBOL Web サービスは、Webサービス用の SOAP サーバー及び Webサーバーとしての機能をもつ COBOL 専用アプリケーションサーバーを使用して展開することができ、COBOL Webサービスを確実に実行するための、スケーラブルなトランザクション環境が提供されます。

また Visual COBOL には、COBOL からの Web サービスの運用のための、Web サービス COBOL クライアントサポートも含まれています。このサポートを使用すれば、標準ベースの WSDL ファイル（.NET や WebSphere で作成された Web サービスなど）またはマッピングファイルから、テンプレートの COBOL クライアントプログラムを構築することができます。

主な機能

Visual Studio を COBOL IDE として利用可能



以下のような多くの COBOL テンプレートを提供

› .NET8 向けのマネージ COBOL テンプレート
- コンソールアプリケーション
- クラスライブラリ

› .NET Framework 4.x 向けのマネージ COBOL テンプレート
- Windows フォームアプリケーション
- Windows フォームコントロールライブラリ
- クラスライブラリ
- コンソールアプリケーション
- Windows サービス
- WCF サービスライブラリ
- WPF アプリケーション/ユーザーコントロールライブラリ
- 配信サービスライブラリ
- SQL Server データベースプロジェクト

› Windows 向けのネイティブ COBOL テンプレート
- Windows アプリケーション
- コンソールアプリケーション
- リンクライブラリ
- Enterprise Server アプリケーション
- ユニットテストライブラリ
- WSDL/JSON から作成する Web サービスクライアントアプリケーション
- Dialog System アプリケーション（最新化／クラシック）

› Web アプリケーション向けの COBOL テンプレート
- ASP.NET Core Web アプリケーション
- ASP.NET Web アプリケーション
- ASP.NET AJAX サーバーコントロール
- ASP.NET サーバーコントロール
- WCF サービスアプリケーション
- WCF REST サービスアプリケーション

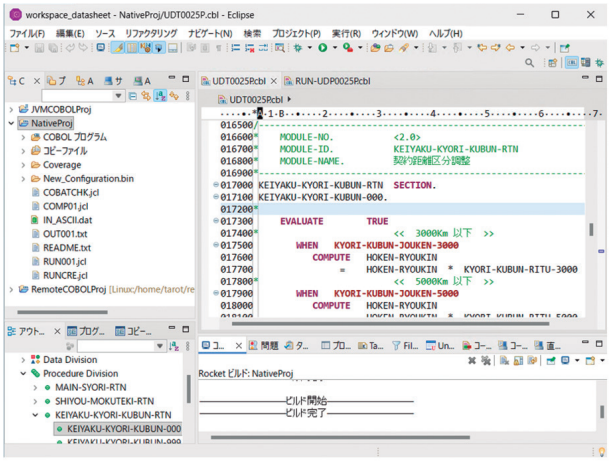
› SQL Server、Microsoft Azure SQL Database/V12 のストアドプロシージャをサポート

› COBOL と C#、VB .NET などの複合アプリケーション開発を実現

› WinForm/WebForm ペインタ、COBOL スニペットもサポート

› アンマネージドコード・アプリケーションのメモリ内の実データを参照できるメモリウィンドウ

Eclipse を COBOL IDE として利用可能



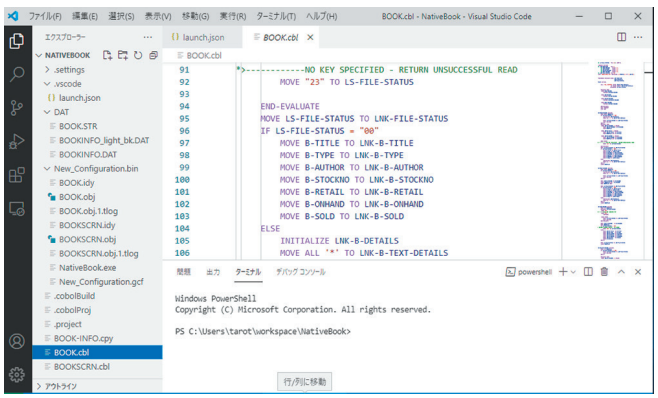
› COBOL パースペクティブ、デバッグパースペクティブなど、該当するペインやビューを IDE に表示するパースペクティブ

› プロジェクト処理
- 開発サーバー上で IDE を使用するローカルプロジェクト
- Windows クライアント上で IDE を使用して Linux/UNIX サーバー上の開発を行うリモートプロジェクト
- 共通 COPY メンバーを管理するコピーブックプロジェクト
- COBOL 専用の単体テストフレームワーク MFUnit と連動したユニットテストプロジェクト
- 自動ビルド
- 従属関係チェック
- エンジンとしての Apache Ant
- ディレクトリに関するビルド優先順位
- SQL 指令サポート
- 複数のビルド構成

Visual Studio Code を COBOL IDE として利用可能

› マーケットプレイスから拡張機能をインストール可能
- 入力支援機能を用いた COBOL 開発
- コンパイル
- ネイティブ、.NET COBOL の実行、デバッグ

コンパイル、実行、デバッグには、ライセンス登録された Visual COBOL 製品が必要です。



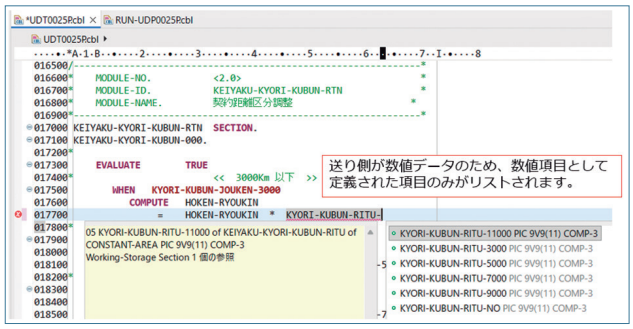
高機能 COBOL エディタ

› バックグラウンド解析による即時の COBOL 構文解析

› Ctrl + Space 打鍵による自動入力

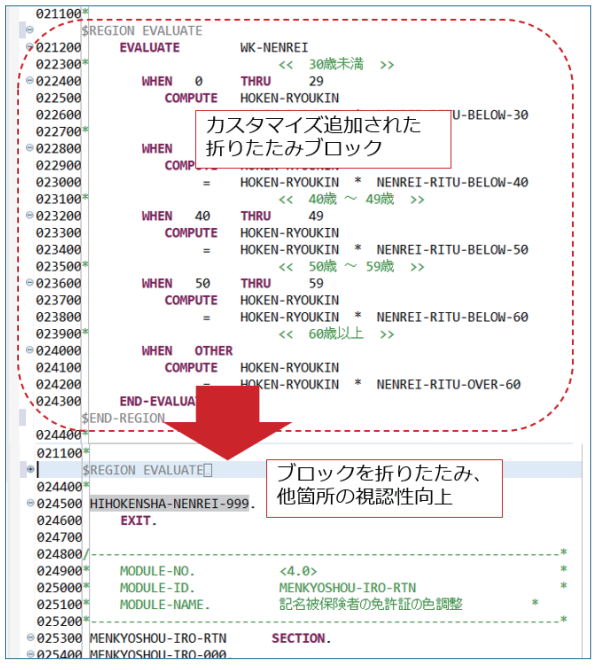
- ビルドインの COBOL コードテンプレート
- カスタム定義した COBOL コードテンプレート
- 定義済みの変数、ファイル名、段落名、節名等
- データの型を認識したリスト表示による誤選択抑止

› 誤入力を自動で補正する AutoCorrect



› 効率的なプログラムアクセス

- アウトラインビュー（Eclipse 版のみ）
- スクロールバー（Visual Studio 版のみ）
- 節・段落毎の折りたたみ・展開
- $REGION による折りたたみ・展開ブロックのカスタマイズ定義



› COBOL 文、変数、コメント等を色分け表示

› 一連番号領域及び見出し領域への自動リナンバリング

› COBOL のコーディングフォーマットと連動したスマート編集モード

› COBOL プログラムソース内でのコピーブックのインライン展開

› コードの可読性を向上させるフォーマッター機能

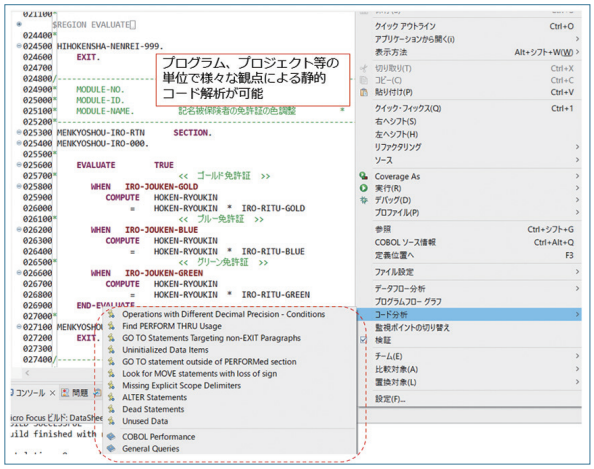
› コピー従属関係を反映したプロジェクトビュー

› 任意のブロックをコピーブックへ外出し

› COBOL 方言に合わせた大文字・小文字変換

› 強力な検索・解析機能

- COBOL ソース情報検索による COBOL 固有の検索機能
- 定義済みの変数、ファイル、段落、節等のクロス参照
- コード解析機能による COBOL の静的コード解析

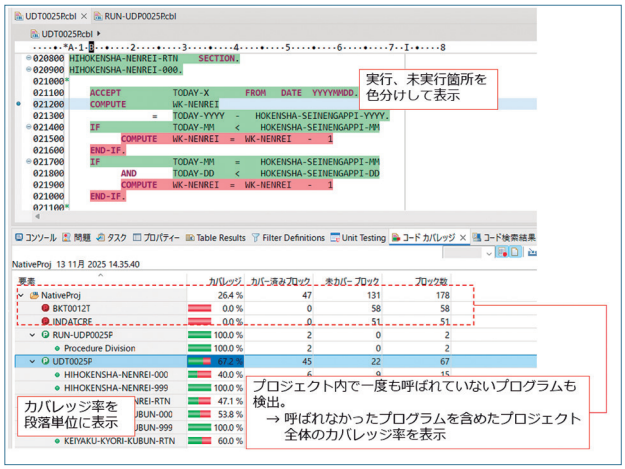


› 節・段落間の関係を俯瞰する呼び出し階層ビュー（Visual Studio 版のみ）

› Data File Tool による IDE 上での順、相対、索引ファイルのメンテナンス

› Visual Studio/Eclipse IDE 上でのカバレッジ分析

- 共通コピーブックの色分け表示



› Visual Studio/Eclipse IDE 上でのパフォーマンス分析

› タブをスペースで置換

大規模プロジェクト対応

› エクスプローラーとアウトラインビューにフィルターを実装

› DLL と EXE の同時リンク

› エディタからファイルプロパティを表示

› ビルドキャンセル機能

› プロジェクトビルド設定の上書き

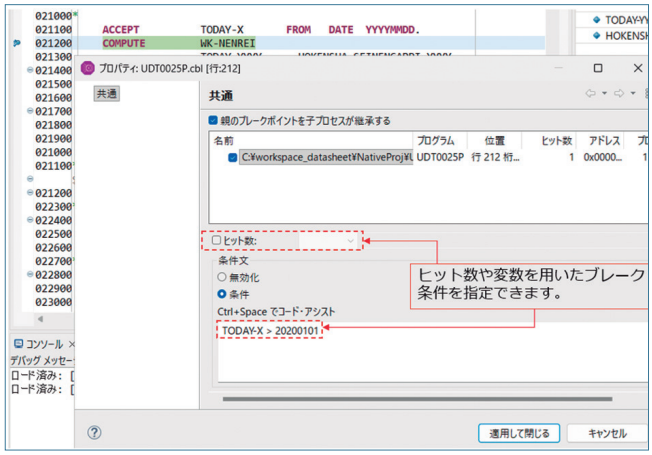
COBOL 専用デバッガーによるテスト効率の向上

› 各デバッグシーンを想定した豊富なデバッグパターンを用意

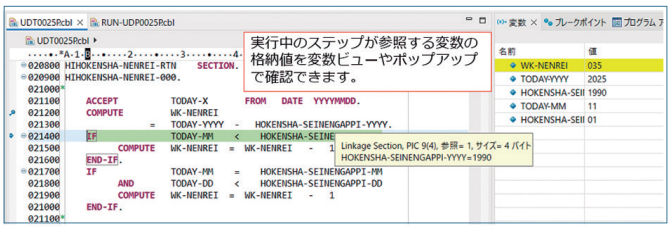
- Windows ネイティブアプリケーションのデバッグ
- .NET マネージ COBOL アプリケーションのデバッグ、他の .NET 言語との混合デバッグ可能（Visual Studio 版のみ）
- JVM COBOL アプリケーションのデバッグ、Java をはじめ JVM 言語との混合デバッグ可能（Eclipse 版のみ）
- Windows ローカルアプリケーションへのアタッチデバッグ
- 特定プロセスへアタッチデバッグ
- Enterprise Server インスタンス上で稼働する COBOL サービスへのアタッチデバッグ
- Linux/UNIX 上の COBOL アプリケーションへのリモートデバッグ（Eclipse 版のみ）
- 異常終了時におけるアプリケーションの状態

› COBOL 向けに作りこまれたデバッグ支援機能

- デバッガーの一時停止位置をコントロールする各種機能
- ステップイン、ステップオーバーによるステップ実行
- ブレークポイントによる任意行でのデバッガーの一時停止
  → ヒットカウント、条件付のブレークポイント、ウォッチポイント設定も可

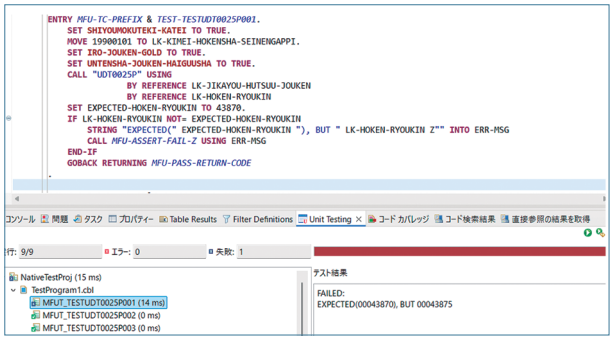


- カーソル位置までステップを実行
- 特定の変数の格納値が変更されるタイミングでデバッガーを一時停止させるウォッチポイント
- テキスト形式や 16 進表示で変数格納値をモニタリング
- ステップ位置の COBOL 文で参照する変数の格納値を参照するビュー
- 予め指定した変数の格納値を参照するビュー
- カーソル位置にある変数の格納値をポップアップ表示



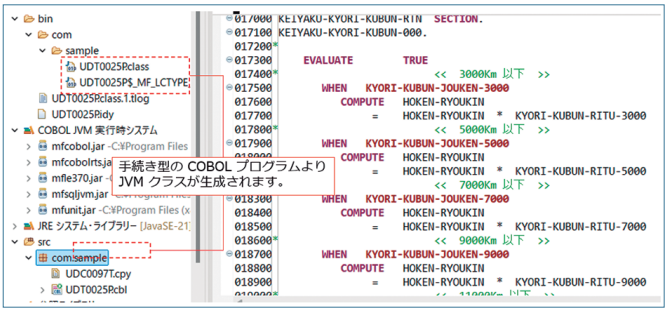
› MFUnit：COBOL 専用の単体テストフレームワーク

- JUnit 同様 xUnit 系の単体テストフレームワーク
- COBOL エントリーポイントのサポート
- ネイティブコード、.NET の CIL コード、Java バイトコードのいずれもサポート
- JUnit 互換形式、テキストファイル形式、マークダウン形式、CSV タイミング形式などの結果ファイルの出力が可能
- コマンドラインによる実行が可能
  →CI ツール等による自動実行、並びに JUnit 向けのツールを活用した結果の集計が可能
- データドリブンテストに対応
- Eclipse、Visual Studio IDE に組み込んだ専用のビューから実行が可能
- テストロジックをテスト対象プログラムに挿入するテストを実施
  テストに必要な前処理など、テスト対象プログラムのロジックをそのまま利用できるため、テスト開発工数を短縮できます。



JVM COBOL

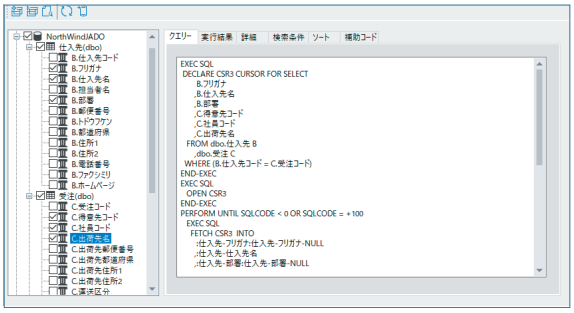
Visual COBOL には Java バイトコードを生成する COBOL コンパイラー「JVM COBOL」が搭載されています。
これにより COBOL ランタイムは Java クラスとして動作し、COBOL アプリケーションは Java のバイトコードとして Java 仮想マシン上で動かすことができるようになり、業界標準のデータセンター運用に準拠したアプリケーション展開を実現します。



強力なデータベースプログラミング機能

› OpenESQL: 埋め込み SQL 文による ODBC/ADO.NET/JDBC データソースへのアクセス

- データソース照会と埋め込み SQL 文を自動作成、テストを効率化



- 共通の埋め込み SQL 文で ODBC、ADO.NET、JDBC のいずれのデータソースへもアクセス可能
- EXEC ADO 構文により、ADO .NET アプリケーションの開発を COBOL で実現
- Pro*COBOL から .NET マネージドへのスムーズな移行を支援する各種機能

› Oracle Pro*COBOL によるアクセス

- COBOL プログラムから、Pro*COBOL によって、Oracle へのアクセスも可能（Oracle が提供する Pro*COBOL が必要）
- COBSQL の利用により Pro*COBOL でプリコンパイルするソースに対してもプリコンパイル展開前のソースを直接 Visual Studio 並びに Eclipse 上で編集・デバッグが可能

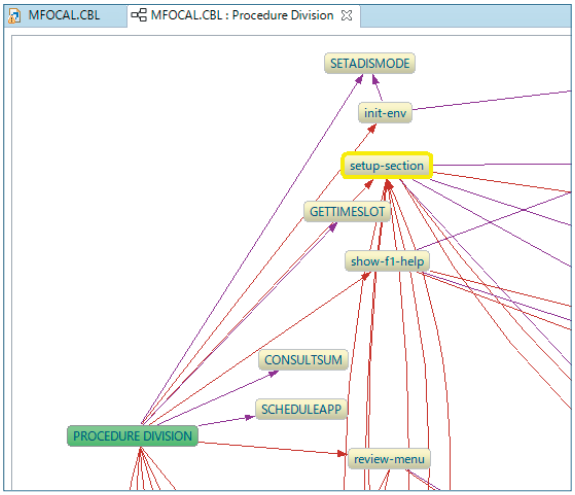
› DB2 ECM による Db2 へのアクセス

- COBOL プログラムから DB2 ECM によって、Db2 へのアクセスが可能。プリコンパイル展開前のソースを直接 Visual Studio 並びに Eclipse 上で編集・デバッグが可能
  (IBM が提供する Db2 COBOL プリコンパイラが必要)

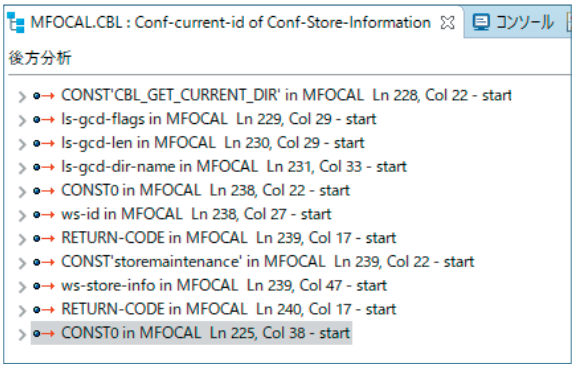
フロー分析機能による COBOL 資産の可視化

既存の COBOL 資産を継続的に活用するためには、資産内容の把握が重要です。フロー分析機能は、プログラム全体の俯瞰や、データ項目の影響範囲を一覧で確認できるようになります。

› プログラムフロー

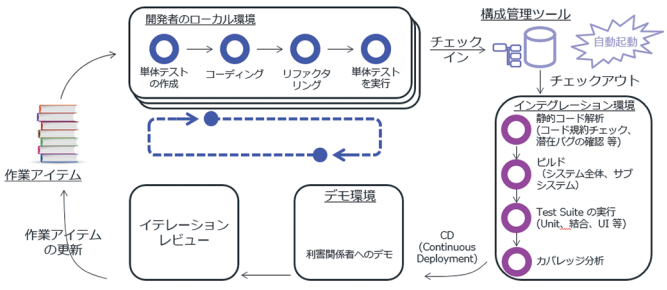


› データフロー



開発プロセスのモダナイズ

› COBOL 開発に継続的インテグレーション（CI）の導入
› CI ツールを利用したエンタープライズ DevOps プロセスを導入し、システムのリリースを高品質、 かつ、 迅速に実現



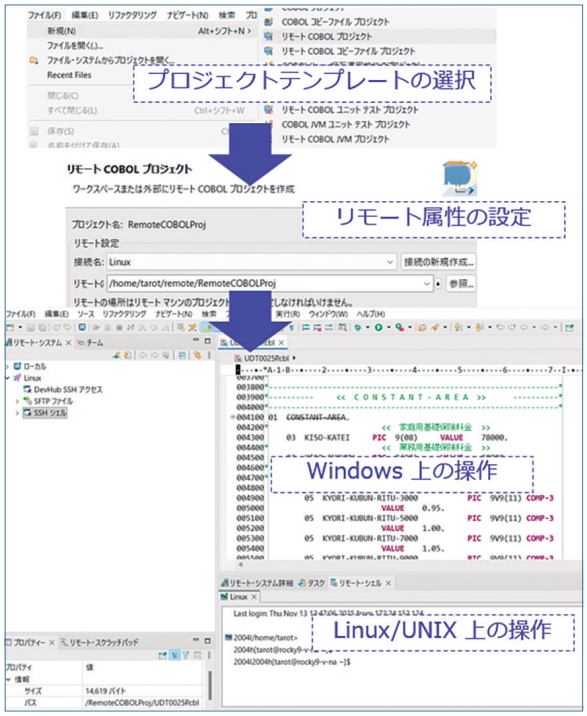
Widows ターゲットの分散開発

› Windows サーバー上の Build Tool による分散開発
- Windows ターゲットのリモートビルド・デバッグ

Linux/UNIX ターゲットの分散開発

› Windows 上の Eclipse による分散開発
- Linux/UNIX ターゲットの COBOL 開発を Windows 上の Eclipse プロジェクトで、実際の開発対象ソースは Linux/UNIX 上で管理
- 接続診断機能

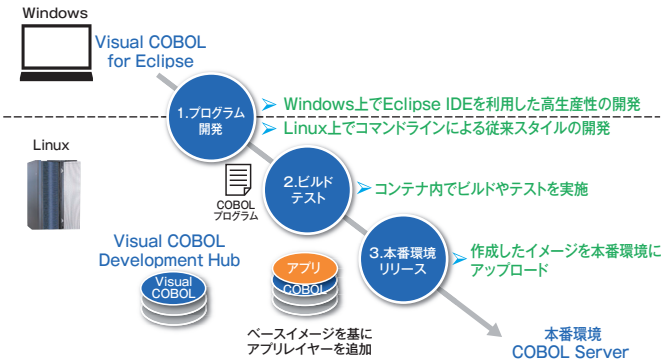
› リモートビルド・デバッグ
- コンパイラ・デバッガも Linux/UNIX 上でバックグラウンド稼働
- Linux/UNIX 上で常駐するリモートデーモンが Eclipse プラグインと通信



Docker/Podman コンテナ利用で開発から運用までを効率化

› 開発
- アプリケーションやライブラリを包括して管理
- 高速に起動し、起動毎に同一環境の構築
  - CI プロセスの実行環境に利用
  - トラブル発生時の再現・調査環境への利用

› 運用
- 検証済み環境のコンテナイメージを本番環境にアップロードで完了
- オーケストレーションソフトと連携し、コンテナ実行環境を効率的に管理



ARM プロセッサ対応

› 運用コストを抑えた環境を選択可能
  Red Hat Enterprise Linux のみ対応

稼働環境

› Windows
› Red Hat Enterprise Linux
› Rocky Linux
› SUSE Enterprise Linux
› Oracle Linux
› Amazon Linux
› AIX

※ Visual COBOL の稼働環境について、最新のサポート状況は弊社ホームページでご確認ください。

※ 開発環境に Visual Studio IDE をご利用の際は、事前に Visual Studio をインストールする必要があります。

Modernization. Without Disruption.™

AMC ソフトウェアジャパン合同会社
https://www.amc.rocketsoftware.co.jp

ASJ-VC-2512-00MR | © 2025 Rocket® Software